2. CIVL-C Language

CIVL-C adds to C a number of concurrency and verification primitives, and also accepts nested
functions. A CIVL-C program consists of a host language file (a C program) and a
verification file (a CIVL-C file).

3. CIVL States and Scopes

CIVL explores a state space which represents all possible states of the concurrency system, taking
into account the possible values of all shared variables. The CIVL state space is
structurally divided into concurrency scopes, which are defined by the
concurrent program.

4. Hierarchical libraries

Implementations of different concurrency libraries have hierarchical structures. The
CIVL core libraries provide concurrency functionalities and concurrency utilities, which
are used by the support libraries of a dialect. The APIs of each dialect are then
implemented upon their own support libraries.

5. Sample Program: Dining Philosophers

Here we present a CIVL-C example which describes the classic dining philosophers problem:

```c
#include "civl-c"

int n = 5; // number of philosophers
int B = 7; // upper bound of n

#assume(expr)

int forks[n];

double x; // x should be unique

void main(int id){
    int left = id; int right = id + 1 % n;
    while (1) {
        #forks(left) = forks(right) = 1;
        if (forks(left) == forks(right)) {
            #forks(left) = forks(right) = 0;
            #x = x + 1;
        }
        #fork(left) = 0;
    }
}
```

6. Transformations

CIVL can be used to perform program transformations. For example, a program that
uses a CIVL-C library function can be transformed into a CIVL-C program and
compiled into a C program.

7. Verification

CIVL explores a state space which represents all possible states of the concurrency system,
taking into account the possible values of all shared variables. The CIVL state space is
structurally divided into concurrency scopes, which are defined by the
concurrent program.

The CIVL-C library provides a number of concurrency and verification primitives that
allow the user to specify the intended behavior of the concurrent system.

The CIVL-C library also provides a number of concurrency and verification primitives that
allow the user to specify the intended behavior of the concurrent system.

The CIVL-C library also provides a number of concurrency and verification primitives that
allow the user to specify the intended behavior of the concurrent system.